

PATENTS

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

**Applicants:** Ryuji Sato                      **Examiner:** Unassigned  
**Serial No.:** To be assigned              **Art Unit:** Unassigned  
**Filed:** Herewith                      **Docket:** 14594  
**For:** PROGRAM CREATION METHOD AND **Dated:** May 1, 2001  
         PROGRAM EXECUTION METHOD

JC872 U.S. PTO  
09/846833  
05/01/01

Assistant Commissioner for Patents  
Washington, DC 20231

**CLAIM OF PRIORITY**

Sir:

Applicant in the above-identified application hereby claims the right of priority in connection with Title 35 U.S.C. §119 and in support thereof, herewith submits a certified copy of Japanese Patent Application No. 2000-136549 filed on May 10, 2000.

Respectfully submitted,

*[Signature]*  
Paul J. Esatto, Jr.  
Registration No. 30,749

SCULLY, SCOTT, MURPHY & PRESSER  
400 Garden City Plaza  
Garden City, New York 11530  
(516) 742-4343

PJE:lf

**CERTIFICATE OF MAILING BY "EXPRESS MAIL"**

"Express Mail" Mailing Label Number: EL798805225  
Date of Deposit: May 1, 2001

I hereby certify that this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. §1.10 on the date indicated above and is addressed to the Assistant Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Dated: May 1, 2001

*[Signature]*  
Janet Grossman

日本国特許庁

PATENT OFFICE  
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application:

2000年 5月10日

出願番号

Application Number:

特願2000-136549

出願人

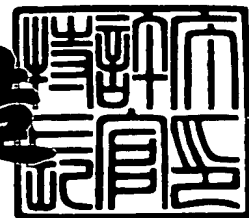
Applicant(s):

日本電気株式会社

2001年 3月 9日

特許庁長官  
Commissioner,  
Patent Office

及川耕造



出証番号 出証特2001-3016643

【書類名】 特許願

【整理番号】 68501827

【提出日】 平成12年 5月10日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 12/00

【発明者】

    【住所又は居所】 東京都港区芝五丁目7番1号 日本電気株式会社内

    【氏名】 佐藤 隆士

【特許出願人】

    【識別番号】 000004237

    【氏名又は名称】 日本電気株式会社

【代理人】

    【識別番号】 100088812

    【弁理士】

    【氏名又は名称】 ▲柳▼川 信

【手数料の表示】

    【予納台帳番号】 030982

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 プログラム構築方法及びそのプログラムの実行方法

【特許請求の範囲】

【請求項 1】 各々対応する処理コードを暗号化した暗号化コードをその実行時に相互に解除する第 1 及び第 2 のプログラムで構築したことを特徴とするプログラム構築方法。

【請求項 2】 前記第 1 及び第 2 のプログラム各々は、前記暗号化コードと前記暗号化コードを解除する復号処理コードとを持つよう構築したことを特徴とする請求項 1 記載のプログラム構築方法。

【請求項 3】 前記第 1 のプログラムは、全体の処理動作において奇数の順序で実行される暗号化コードを含み、

前記第 2 のプログラムは、前記全体の処理動作において偶数の順序で実行される暗号化コードを含むことを特徴とする請求項 2 記載のプログラム構築方法。

【請求項 4】 前記第 1 及び第 2 のプログラム各々は、前記暗号化コードを解除する復号処理コードを持ち、

前記暗号化コードは、前記第 1 及び第 2 のプログラム各々の共有領域に含むよう構築したことを特徴とする請求項 1 記載のプログラム構築方法。

【請求項 5】 前記復号処理コードは、前記処理コードに含むよう構築したことを特徴とする請求項 1 から請求項 4 のいずれか記載のプログラム構築方法。

【請求項 6】 前記第 1 及び第 2 のプログラム各々において暗号解除を待つ側は、予め算出した算出時間だけ待機し、前記算出時間が経過すると前記暗号解除の有無にかかわらず次の処理を実行するよう構築したことを特徴とする請求項 1 から請求項 5 のいずれか記載のプログラム構築方法。

【請求項 7】 前記第 1 及び第 2 のプログラム各々は、ソフトウェアデバガによる動的な解析で前記暗号化コードの相互解除に遅れが生じた時に異常終了するよう構築したことを特徴とする請求項 1 から請求項 6 のいずれか記載のプログラム構築方法。

【請求項 8】 前記暗号化コードは、前記第 1 及び第 2 のプログラムの両方

で復号しないと元の処理コードに戻らないよう構築したことを特徴とする請求項 1 から請求項 7 のいずれか記載のプログラム構築方法。

【請求項 9】 前記第 1 及び第 2 のプログラムの起動時に前記第 1 及び第 2 のプログラムによって 1 番目の暗号化コードを復号してから前記第 1 のプログラムによってその復号された 1 番目の処理コードを実行し、前記第 1 及び第 2 のプログラムによって 2 番目の暗号化コードを復号してから前記第 2 のプログラムによってその復号された 2 番目の処理コードを実行するよう構築したことを特徴とする請求項 1 から請求項 8 のいずれか記載のプログラム構築方法。

【請求項 10】 第 1 及び第 2 のプログラムの実行時に前記第 1 及び第 2 のプログラム各々に対応する処理コードを暗号化した暗号化コードを相互に解除するようにしたことを特徴とするプログラム実行方法。

【請求項 11】 前記第 1 及び第 2 のプログラム各々は、前記暗号化コードと前記暗号化コードを解除する復号処理コードとを持つようにしたことを特徴とする請求項 10 記載のプログラム実行方法。

【請求項 12】 前記第 1 のプログラムは、全体の処理動作において奇数の順序で実行される暗号化コードを含み、

前記第 2 のプログラムは、前記全体の処理動作において偶数の順序で実行される暗号化コードを含むようにしたことを特徴とする請求項 11 記載のプログラム実行方法。

【請求項 13】 前記第 1 及び第 2 のプログラム各々は、前記暗号化コードを解除する復号処理コードを持ち、

前記暗号化コードは、前記第 1 及び第 2 のプログラム各々の共有領域に含むようにしたことを特徴とする請求項 10 記載のプログラム実行方法。

【請求項 14】 前記復号処理コードは、前記処理コードに含むようにしたことを特徴とする請求項 10 から請求項 13 のいずれか記載のプログラム実行方法。

【請求項 15】 前記第 1 及び第 2 のプログラム各々において暗号解除を待つ側は、予め算出した算出時間だけ待機し、前記算出時間が経過すると前記暗号解除の有無にかかわらず次の処理を実行するよう構築したことを特徴とする請求

項 1 0 から請求項 1 4 のいずれか記載のプログラム実行方法。

【請求項 1 6】 前記第 1 及び第 2 のプログラムは、ソフトウェアデバッガによる動的な解析で前記暗号化コードの相互解除に遅れが生じた時に異常終了するようにしたことを特徴とする請求項 1 0 から請求項 1 5 のいずれか記載のプログラム実行方法。

【請求項 1 7】 前記暗号化コードは、前記第 1 及び第 2 のプログラムの両方で復号しないと元の処理コードに戻らないようにしたことを特徴とする請求項 1 0 から請求項 1 6 のいずれか記載のプログラム実行方法。

【請求項 1 8】 前記第 1 及び第 2 のプログラムの起動時に前記第 1 及び第 2 のプログラムによって 1 番目の暗号化コードを復号してから前記第 1 のプログラムによってその復号された 1 番目の処理コードを実行し、前記第 1 及び第 2 のプログラムによって 2 番目の暗号化コードを復号してから前記第 2 のプログラムによってその復号された 2 番目の処理コードを実行するようにしたことを特徴とする請求項 1 0 から請求項 1 7 のいずれか記載のプログラム実行方法。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明はプログラム構築方法及びそのプログラムの実行方法に関し、特にソフトウェアの動的な解析を阻止するための技術に関する。

【 0 0 0 2 】

【従来の技術】

従来、ソフトウェアに関する不正利用への対抗技術としては、例えば「逆解析や改変からソフトを守る－タンパレジスタントソフトウェア技術－」（1998 年 1 月、日経エレクトロニクス通巻 7 0 6 号，p p . 2 0 9 ～ 2 2 0 ）に記載されているように、第三者が秘密性の高いプログラムを解析し、不正利用することを防ぐために用いられている。

【 0 0 0 3 】

その中でも、実行中のプログラムに対してソフトウェアデバッガ等を用いることによって接触し、プログラムを一文ずつ実行させて、その挙動を調べるやりか

たは、完全な回避方法のない強力な解析手段である。

【 0 0 0 4 】

上記の文献に記載されている不正利用への対抗技術における動的な解析への対抗手段について説明する。その構成を図 8 に示す。この図 8 において、プログラム 1 0, 2 0 は不正利用への対抗技術を適用して保護すべきプログラムであり、プログラム 6 0 は不正利用への対抗技術のために用意した監視プログラムである。尚、プログラム 1 0, 2 0, 6 0 は自分自身の改ざんを検出する改ざん検出コード 3 2 をそれぞれ含んでいる。

【 0 0 0 5 】

これらの認証について説明する。プログラム 1 0 とプログラム 6 0 とはデジタル署名による通信プロトコルによって互いの改ざん検出コード 3 2 を認証する。その際、改ざん検出コード 3 2 が壊れたり、改ざんされている場合にはそこで処理を終了する。

【 0 0 0 6 】

プログラム 2 0 とプログラム 6 0 とはデジタル署名による通信プロトコルによって互いの改ざん検出コード 3 3 を認証する。その際、改ざん検出コード 3 2 が壊れたり、改ざんされている場合にはそこで処理を終了する。

【 0 0 0 7 】

これらの方法によって、改ざん検出コード 3 2 が動的な解析を検出した場合には処理を中断することができる。また、2 つの組と相互に認証を行うことで、通信プロトコルのメッセージを模倣してなりすます第三者のプログラムにも対応することができる。

【 0 0 0 8 】

【発明が解決しようとする課題】

上述した従来のソフトウェアに関する不正利用への対抗技術では、ソフトウェアデバッガによる解析でトレースのみを行っている場合、改ざん検出コードがプログラムの改ざんを検出するのみであるので、デバッガ側がプログラムのどこかを改ざんしなくてはその改ざんを検出することができないという問題がある。

【 0 0 0 9 】

そこで、本発明の目的は上記の問題点を解消し、ソフトウェアデバッガによる動的なソフトウェア解析を不可能にすることができるプログラム構築方法及びそのプログラムの実行方法を提供することにある。

## 【 0 0 1 0 】

## 【課題を解決するための手段】

本発明によるプログラム構築方法は、各々対応する処理コードを暗号化した暗号化コードをその実行時に相互に解除する第 1 及び第 2 のプログラムで構築している。

## 【 0 0 1 1 】

本発明によるプログラム実行方法は、第 1 及び第 2 のプログラムの実行時に前記第 1 及び第 2 のプログラム各々に対応する処理コードを暗号化した暗号化コードを相互に解除するようにしている。

## 【 0 0 1 2 】

すなわち、本発明のプログラム構築方法は、2 つのプログラムがそれぞれ暗号化したプログラムコードを持ち、実行時に暗号化したプログラムコードを相互に解除していくことで、ソフトウェアデバッガによるプログラムの動的な解析を妨ぐことができるように構築している。

## 【 0 0 1 3 】

より具体的に、本発明のプログラム構築方法では、第 1 のプログラムが全体の処理動作において奇数の順序で実行される暗号化コードを含み、第 2 のプログラムが全体の処理動作において偶数の順序で実行される暗号化コードを含んでいる。第 1 及び第 2 のプログラムにおいては各々の暗号化コードを互いに交互に実行するように組まれている。暗号化コードはどれも第 1 及び第 2 のプログラムの両方で復号しないと元のコードに戻らないようにしておく。

## 【 0 0 1 4 】

第 1 及び第 2 のプログラムは起動されると、まず 1 番目の暗号化コードを復号し、第 1 のプログラムはその復号されたコードを実行する。次に、第 1 及び第 2 のプログラムは 2 番目の暗号化コードを復号し、第 2 のプログラムはその復号されたコードを実行する。



## 【 0 0 1 5 】

上記の処理を繰り返すことで、ソフトウェアデバッガによって第1のプログラムに動的な解析が行われた場合、暗号化コードが完全に復号されなくなり、それを実行しようとした第2のプログラムは不正なコードを実行したことによって終了し、以後の暗号化コードが完全に復号されなくなった第1のプログラムも終了することになる。

## 【 0 0 1 6 】

本発明では、ソフトウェアデバッガが起動しているひとつのプログラムしか解析することができない点と、動的な解析が通常にプログラムが実行されるよりも時間がかかるという点の2つの特徴を利用している。これによって、不正な動的解析を妨げる構造が可能となる。

## 【 0 0 1 7 】

## 【発明の実施の形態】

次に、本発明の実施例について図面を参照して説明する。図1は本発明の一実施例によるプログラムの構造を示す図である。図1において、プログラム10は暗号化コード11, 13, 15と、暗号化コード12, 14, 16の解除コード12a, 14a, 16aと、復号処理コード30とを含んでおり、プログラム20は暗号化コード12, 14, 16と、暗号化コード11, 13, 15の解除コード11a, 13a, 15aと、復号処理コード31とを含んでいる。

## 【 0 0 1 8 】

図2は図1の暗号化をする前のプログラムを示す図である。図2において、暗号化コード11～16はそれぞれ任意の処理コード1～6を暗号化したコードであり、どれも復号処理コード30及び復号処理コード31で復号しなければ元の処理を行う処理コード1～6に戻らない。

## 【 0 0 1 9 】

その場合、復号処理コード30は解除コード12a, 14a, 16aでプログラム20の暗号化コード12, 14, 16の解除を行い、復号処理コード31は解除コード11a, 13a, 15aでプログラム10の暗号化コード11, 13, 15の解除を行う。

## 【 0 0 2 0 】

図 3 は図 1 のプログラムが実行される順番を示す図である。図 3 においては、プログラム 1 0, 2 0 で時間的にそれぞれ暗号化コード 1 1, 1 2, 1 3, 1 4, 1 5, 1 6 の順番で実行するように構成されていることを示している。

## 【 0 0 2 1 】

異なるプログラムにおいて順番に実行される暗号化コードを構築するには、システムタイマ等の同期機構を利用してプログラムを構築する方法や、初期設定処理、画面描画処理、終了処理等の実行される順番が決まっている処理に、暗号化コードを組込む等の方法があるが、ソフトウェアデバッガによる解析で停止をしない仕組みでなければならない。実際にはシステムタイマ等を用いた時間によるウェイトが望ましい。

## 【 0 0 2 2 】

図 4 は図 1 のプログラムが正しく動作している場合を示す図であり、図 5 は図 1 のプログラムの動作を示すフローチャートである。これら図 1 と図 4 と図 5 とを参照して本発明の一実施例の全体の動作について説明する。

## 【 0 0 2 3 】

まず、プログラム 1 0 とプログラム 2 0 とが起動されると、復号処理コード 3 0 による解除と、解除コード 1 1 a を用いた復号処理コード 3 1 による解除とによって暗号化コード 1 1 の暗号が解除されて処理コード 1 となる（図 5 ステップ S 1, S 1 1）。

## 【 0 0 2 4 】

この時、暗号解除を待つ側はウェイト機構で待機する（図 5 ステップ S 2）。このウェイト機構は並立して動作する 2 つのプログラム 1 0, 2 0 において、同期をとるためのルーチンである。プログラム 2 0 による暗号化コード 1 1 の解除が行われるまで、予め算出した時間だけ待機する。その算出時間がすぎると、暗号化解除の有無にかかわらず、次のステップを実行する。

## 【 0 0 2 5 】

この後に、プログラム 1 0 によって処理コード 1 が実行される（図 5 ステップ S 3）。続いて、解除コード 1 2 a を用いた復号処理コード 3 0 による解除と復

号処理コード31の解除とによって暗号化コード12が解除されて処理コード2となる（図5ステップS4，S12）。

【0026】

この時、暗号解除を待つ側はウェイト機構で待機し（図5ステップS13）、その後に、プログラム20によって処理コード2が実行される（図5ステップS14）。続いて、復号処理コード30による解除と、解除コード13aを用いた復号処理コード31による解除とによって暗号化コード13が解除されて処理コード3となる（図5ステップS5，S15）。

【0027】

この時、暗号解除を待つ側はウェイト機構で待機し（図5ステップS6）、その後に、プログラム10によって処理コード3が実行される（図5ステップS7）。続いて、解除コード14aを用いた復号処理コード30による解除と復号処理コード31による解除とによって暗号化コード14が解除されて処理コード4となる（図5ステップS8，S16）。

【0028】

この時、暗号解除を待つ側はウェイト機構で待機し（図5ステップS17）、その後に、プログラム20によって処理コード4が実行される。というように、プログラム10，20の処理が続けられていく。

【0029】

この手続きによって、プログラム10とプログラム20とのどちらかがデバッグ（図示せず）によって動的なデバッグが行われても、これらプログラム10，20の解析を防ぐことができる。

【0030】

図6は図1のプログラムが動的にデバッグされている場合を示す図である。図6においては、例えばプログラム20が実行中にデバッグ40によって動的に解析されそうになった場合を示している。

【0031】

この場合、プログラム10とプログラム20とが起動されて処理を始めると、復号処理コード30，31によって暗号化コード11が解除されて処理コード1

になり、プログラム 1 0 によって処理コード 1 が実行される。

【 0 0 3 2 】

デバッガ 4 0 によってプログラム 2 0 はデバッグされると、プログラム 2 0 の実行は一時停止するか、デバッガ 4 0 の使用者によってステップ実行され、ゆっくりとした処理速度になる。

【 0 0 3 3 】

復号処理コード 3 0 によって暗号化コード 1 2 は一部の暗号が解除されるとともに（不完全な処理コード 2）、復号処理コード 3 0 によって暗号化コード 1 3 も一部の暗号が解除される（不完全な処理コード 3）。プログラム 1 0 によって不完全な処理コード 3 が実行されると、不正な処理を行ったとして、オペレーティングシステムによってプログラム 1 0 が終了させられる。

【 0 0 3 4 】

復号処理コード 3 1 によって不完全な処理コード 2 は処理コード 2 になり、プログラム 2 0 はその処理コード 2 を実行する。また、復号処理コード 3 1 によって暗号化コード 1 4 は一部の暗号が解除され（不完全な処理コード 4）、プログラム 2 0 によってその不完全な処理コード 4 が実行されると、不正な処理を行ったとして、オペレーティングシステムによってプログラム 2 0 が終了させられる。

。

【 0 0 3 5 】

上記のように、プログラム 2 0 がデバッガ 4 0 のデバッグによって、処理の流れが図 3 の流れよりも遅れると、プログラム 1 0 は一部しか解除されない処理コードを実行し、不正な処理によって終了する。それによって、プログラム 2 0 も一部しか解除されない処理コードを実行し、不正な処理によって終了する。デバッガ 4 0 の解析もそこで終了することになる。

【 0 0 3 6 】

このように、2 つのプログラム 1 0、2 0 が互いのプログラムに対して、段階的に暗号解除処理を行うことで、どちらのプログラムがデバッグされても、暗号解除処理の手順がずれて、不正な処理によって終了するという構造を持つことによって、ソフトウェアデバッガ 4 0 による動的なソフトウェア解析を不可能にす

ることができる。

【0037】

尚、上記の説明では2個のプログラム10、20について述べているが、それらのプログラムは2個ではなく、N個（Nは3以上の整数）でもかまわない。その場合、プログラムをN個に増やすと、プログラム10とプログラム20とが相互の暗号解除を行い、プログラムN-1とプログラムNとが相互の暗号解除を行い、プログラムNとプログラム10とが相互の暗号解除を行うという形になる。これによって、全体的にさらに解析しにくいプログラムとなる。

【0038】

また、図1及び図2において、復号処理コード30、31は処理コード1～6に含めてもかまわない。このように、任意の処理コード1～6に復号処理コード30、31を含めると、第三者による解析の危険をさらに減らすことができる。

【0039】

さらに、図5に示すフローチャートにおいて、ウェイト機構は時間による待機以外にも、第三のプログラムによる同期処理でもかまわない。この場合、このプログラムはプログラム10とプログラム20とに暗号解除を許可するメッセージを順番に送信する。例えば、ステップS11の終了を確認したら、ステップS2で待機しているプログラム10にウェイト終了のメッセージを送信する。

【0040】

この方法を使用した場合にも、ウェイト機構に、一定時間メッセージを受信しなかったら、次のステップに進む処理を入れることで、第三のプログラムへの動的な解析に対抗することができる。

【0041】

図7は本発明の他の実施例によるプログラムの構造を示す図である。図7において、本発明の他の実施例によるプログラム10、20にはそれぞれ復号処理コード30、31を含んでおり、相互に解除する暗号化コード11～13はプログラム10、20上ではなく、別に配設した共有領域50上にある。

【0042】

図7に示すように、プログラム10、20は共有領域50にある暗号化コード

1 1 ~ 1 3 を解除し、それぞれの処理コードを実行していく。オペレーティングシステムの仕様で相手のプログラム上の暗号化コードの書き換えが困難な場合には、この方法によって上記と同様の効果が得られる。

【 0 0 4 3 】

【発明の効果】

以上説明したように本発明によれば、各々対応する処理コードを暗号化した暗号化コードをその実行時に相互に解除する第 1 及び第 2 のプログラムで構築することによって、ソフトウェアデバッガによる動的なソフトウェア解析を不可能にすることができるという効果がある。

【図面の簡単な説明】

【図 1】

本発明の一実施例によるプログラムの構造を示す図である。

【図 2】

図 1 の暗号化をする前のプログラムを示す図である。

【図 3】

図 1 のプログラムが実行される順番を示す図である。

【図 4】

図 1 のプログラムが正しく動作している場合を示す図である。

【図 5】

図 1 のプログラムの動作を示すフローチャートである。

【図 6】

図 1 のプログラムが動的にデバッグされている場合を示す図である。

【図 7】

本発明の他の実施例によるプログラムの構造を示す図である。

【図 8】

従来の動的解析防止方法を示す図である。

【符号の説明】

1 ~ 6    処理コード

1 1 ~ 1 6    暗号化コード

1 1 a, 1 2 a,

1 3 a, 1 4 a,

1 5 a, 1 6 a 解除コード

1 0, 2 0 プログラム

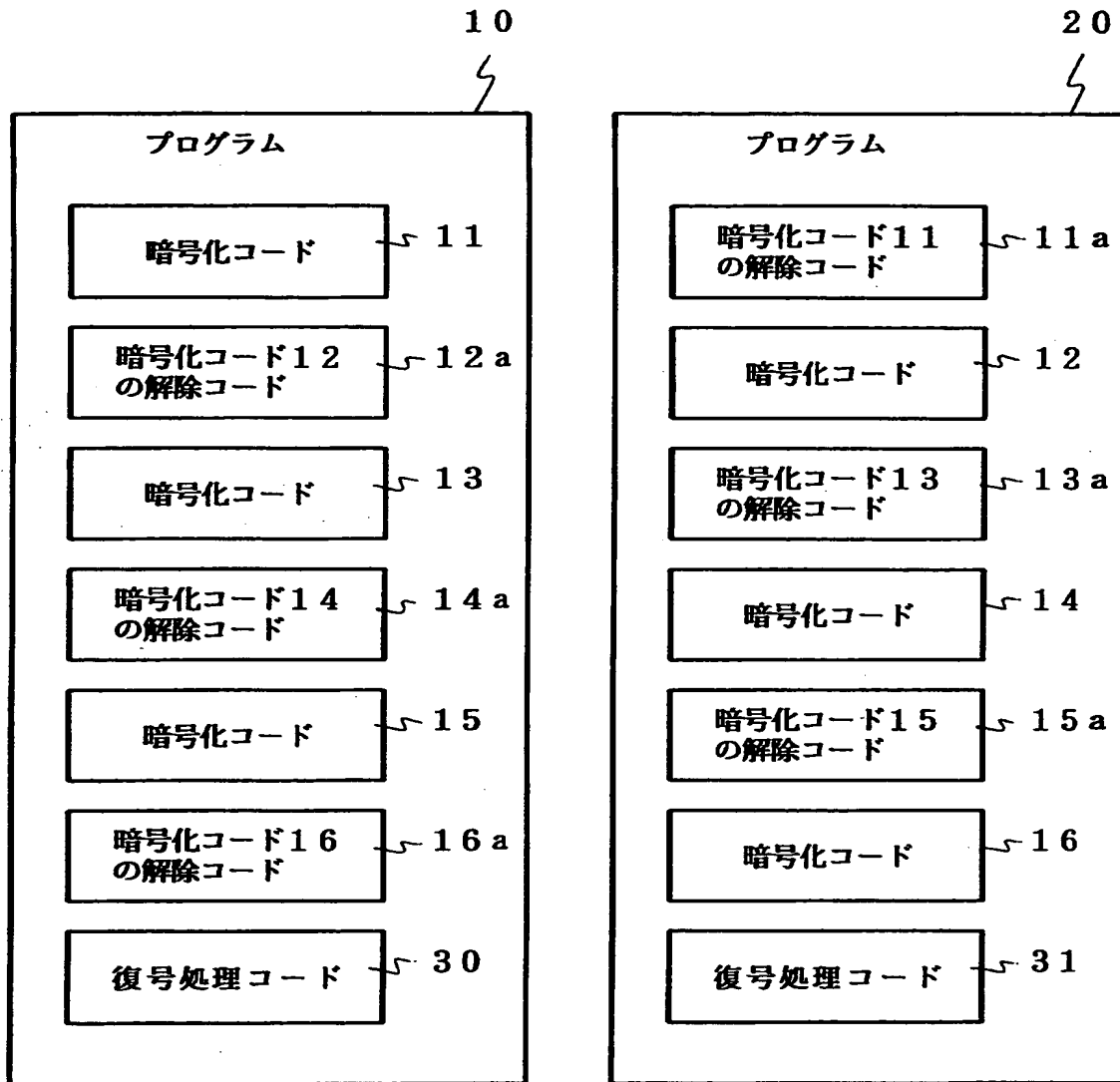
3 0, 3 1 復号処理コード

4 0 デバッガ

5 0 共有領域

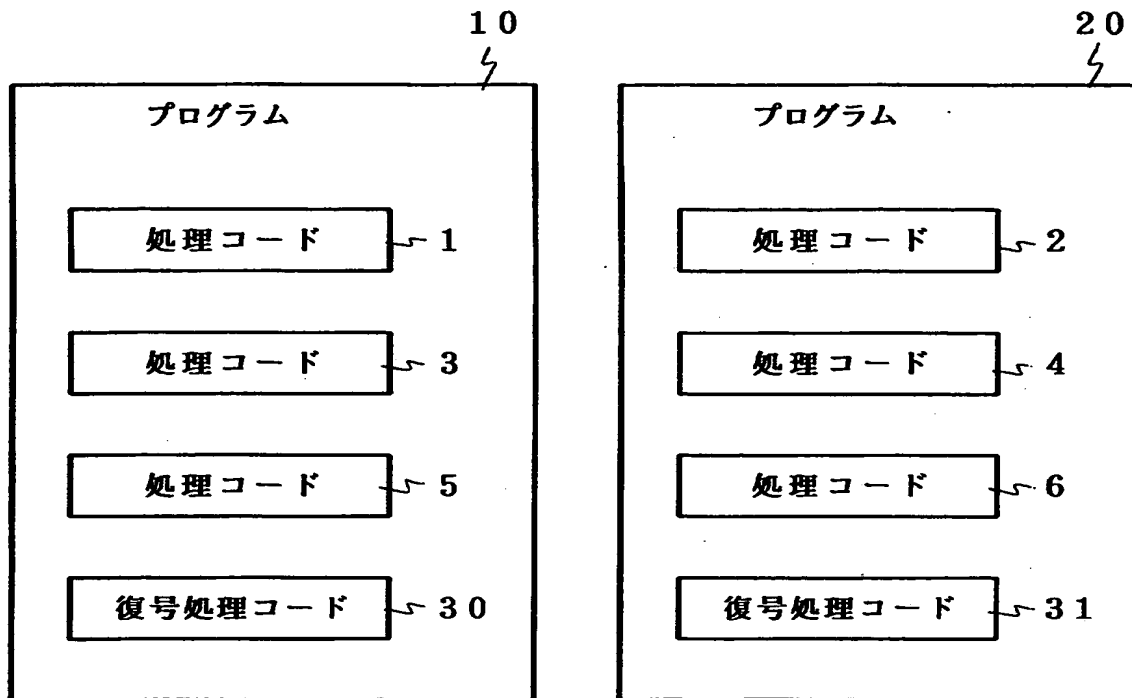
【書類名】 図面

【図 1】



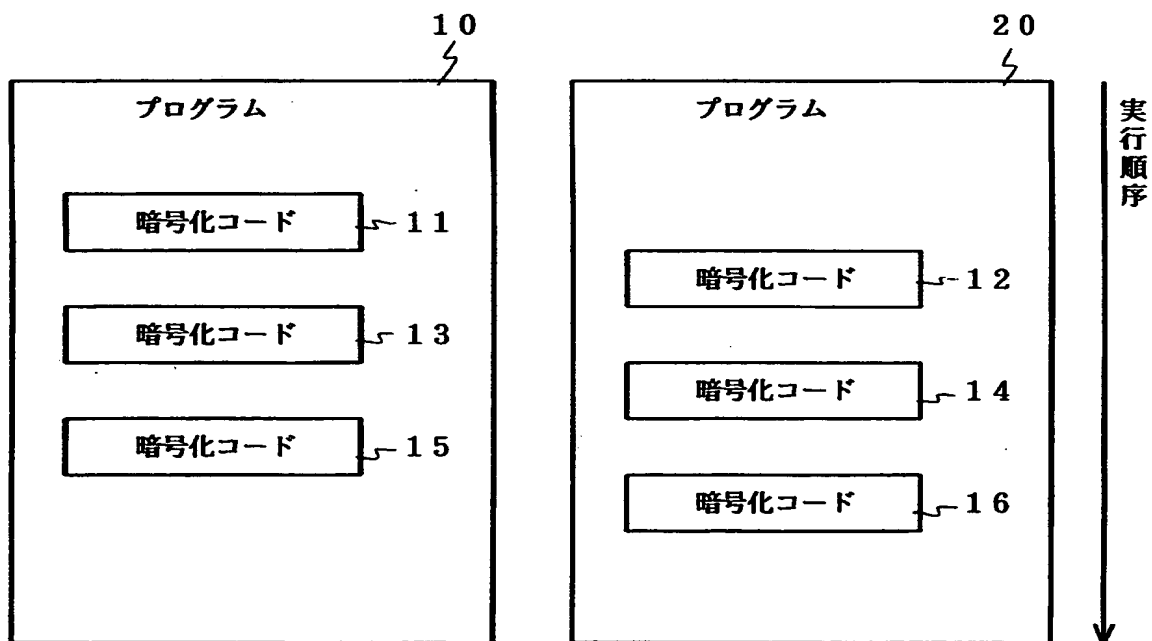


【図 2】

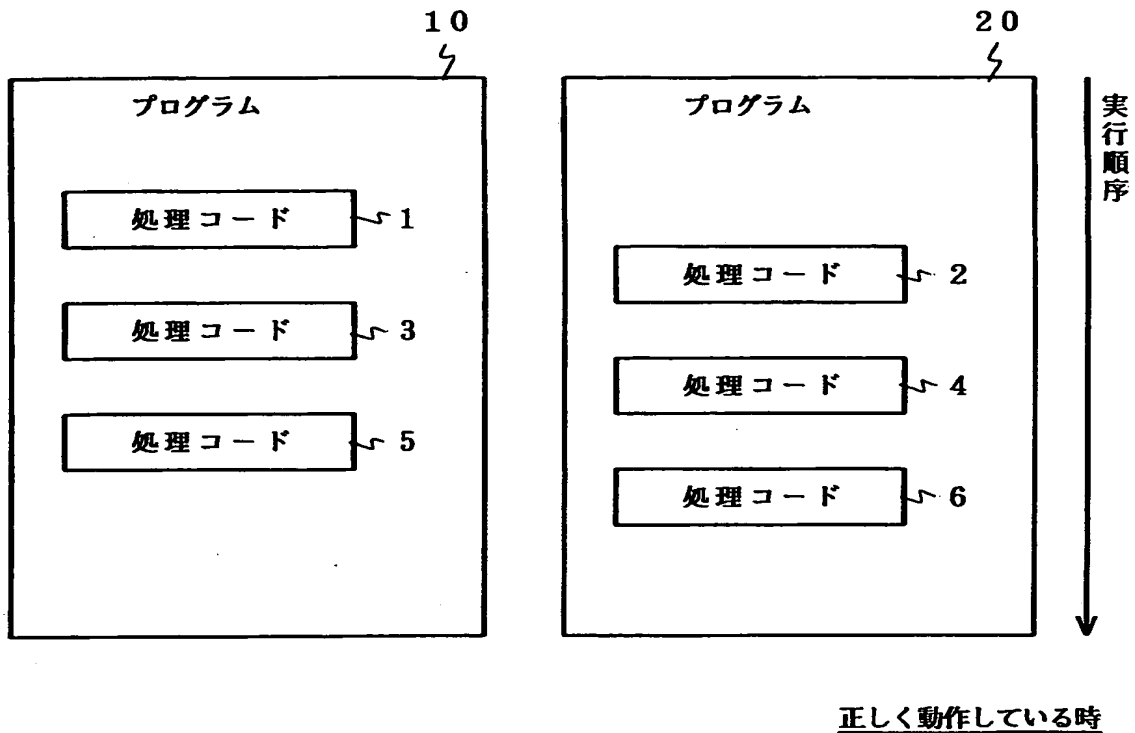


暗号化前の状態

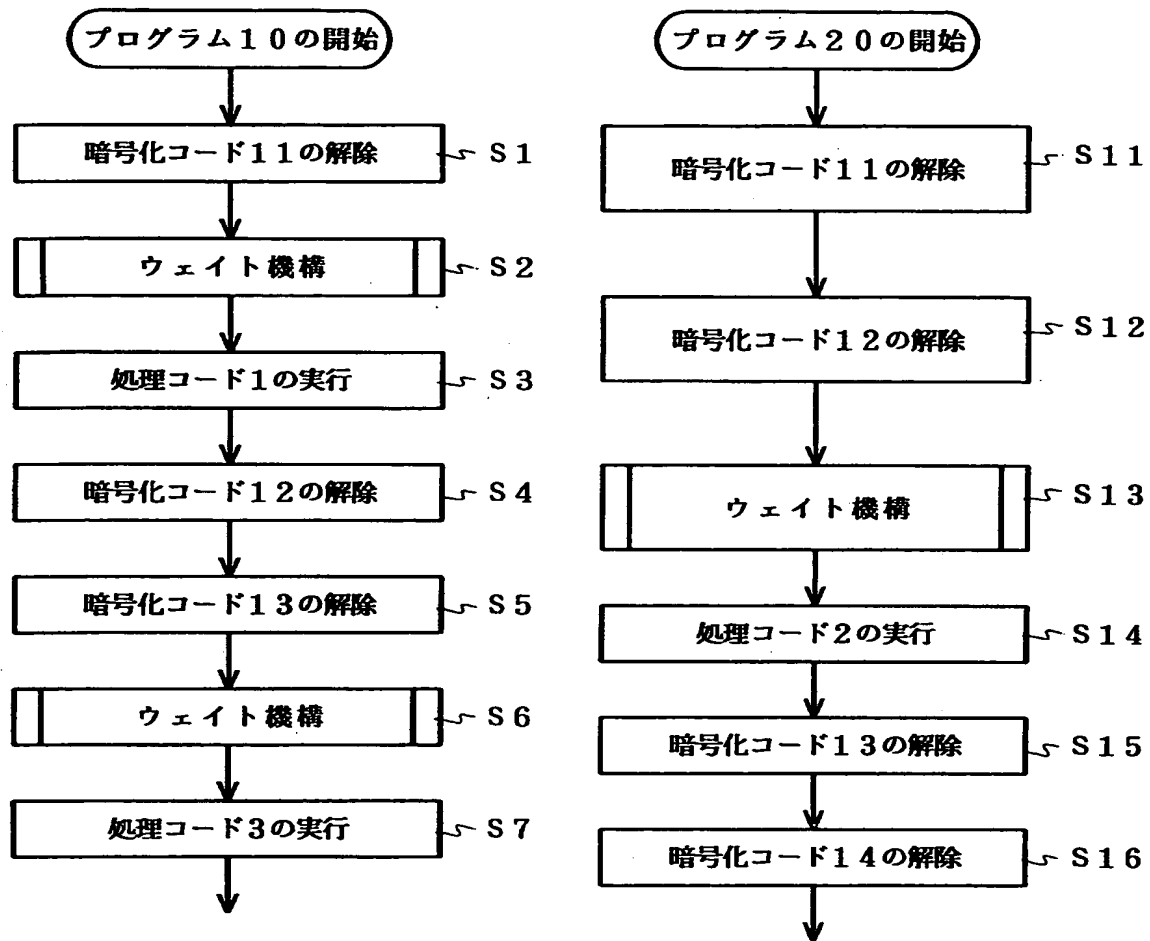
【図 3】



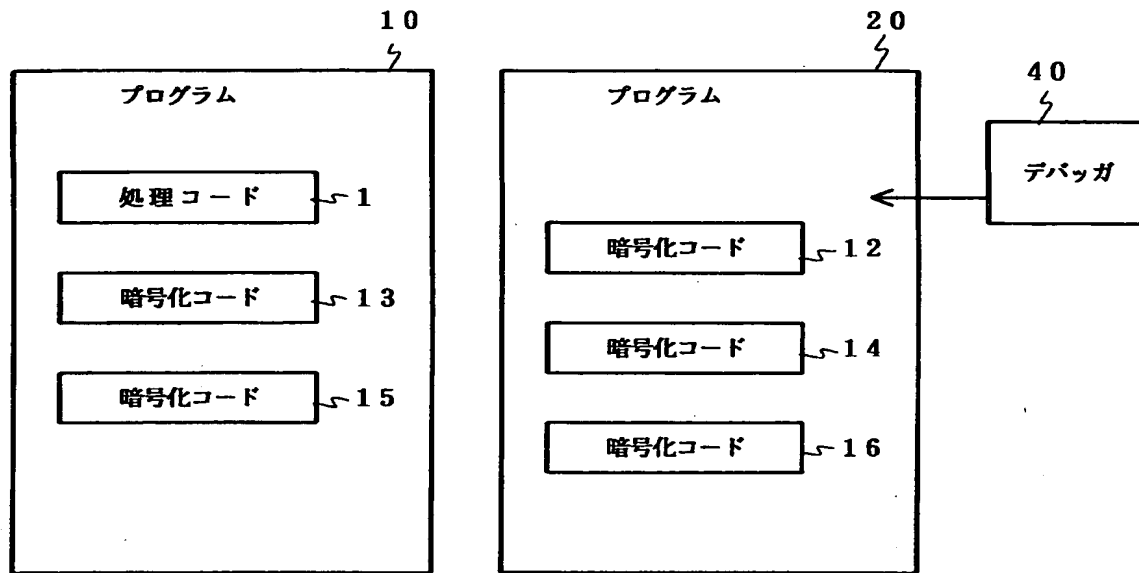
【図 4】



【図 5】

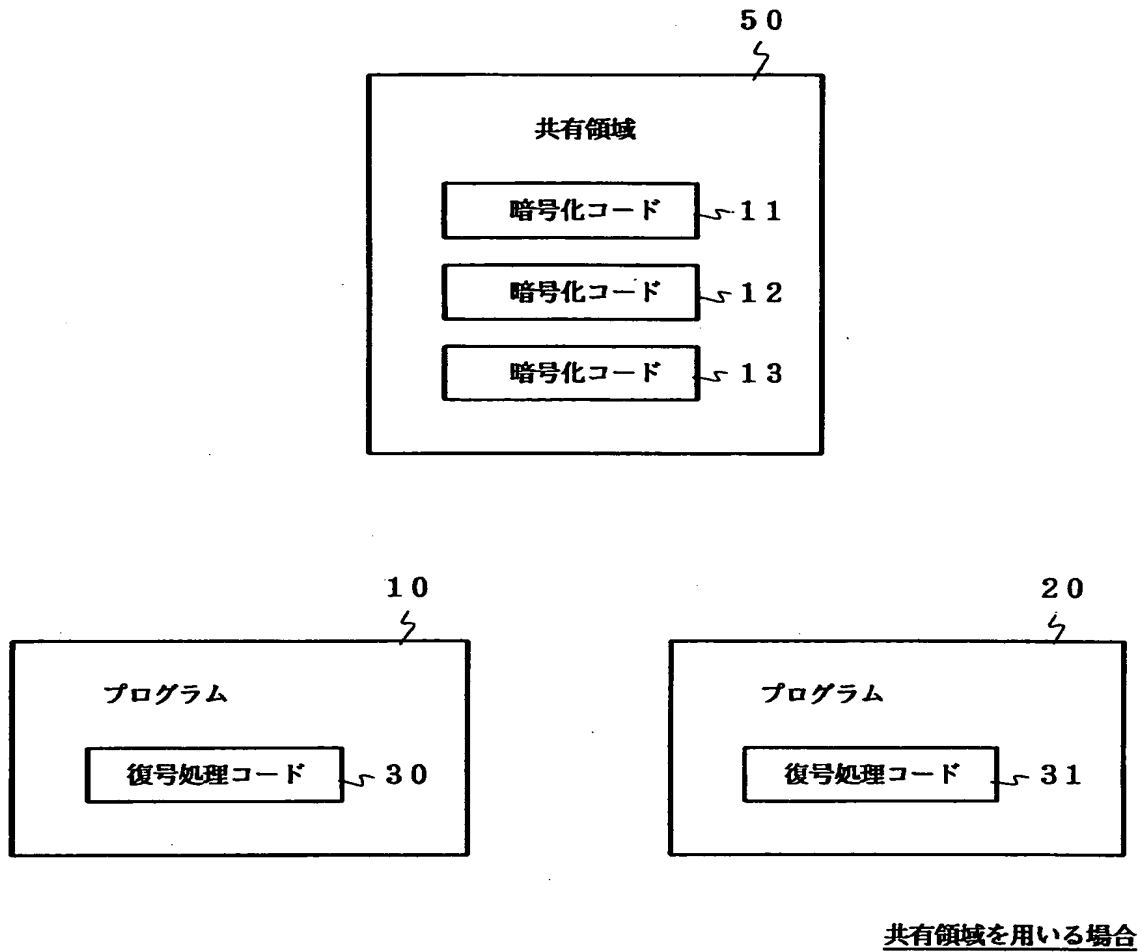


【図 6】

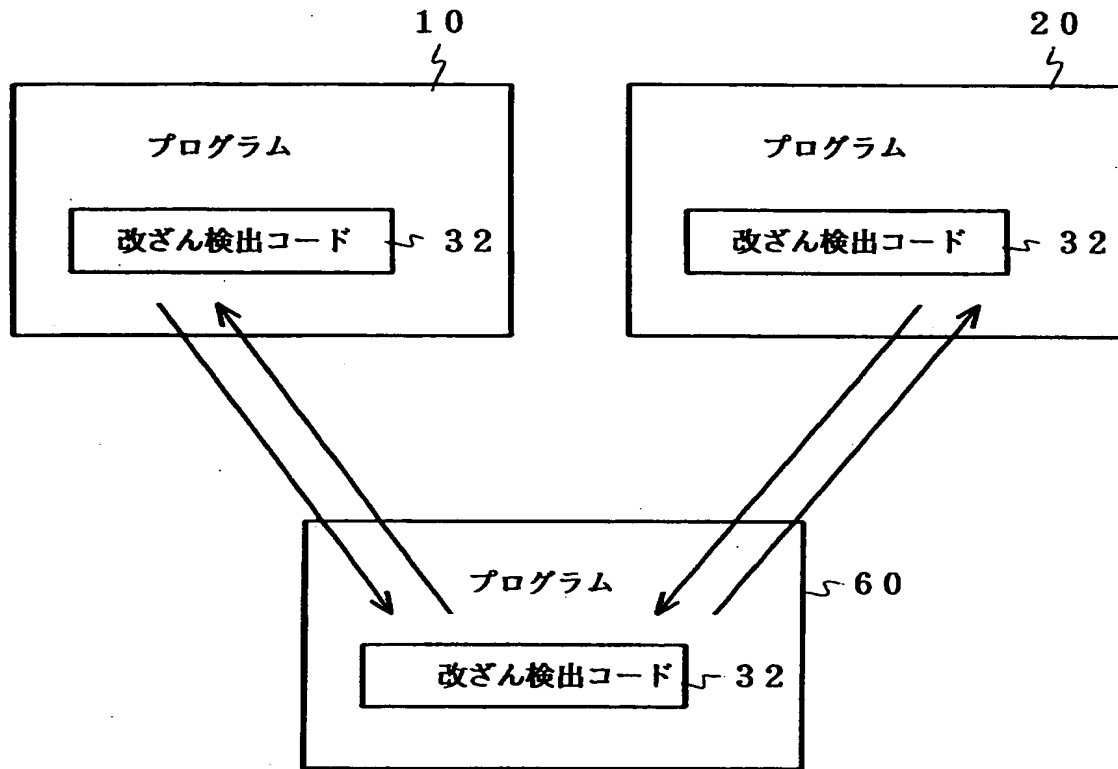


動的にデバッグされた時

【図 7】



【図 8】



【書類名】 要約書

【要約】

【課題】 ソフトウェアデバッガによる動的なソフトウェア解析を不可能にすることが可能なプログラム構築方法を提供する。

【解決手段】 プログラム 1 0 は暗号化コード 1 1, 1 3, 1 5 と、暗号化コード 1 2, 1 4, 1 6 の解除コード 1 2 a, 1 4 a, 1 6 a と、復号処理コード 3 0 とを含んでおり、プログラム 2 0 は暗号化コード 1 2, 1 4, 1 6 と、暗号化コード 1 1, 1 3, 1 5 の解除コード 1 1 a, 1 3 a, 1 5 a と、復号処理コード 3 1 とを含んでいる。暗号化コード 1 1 ~ 1 6 はそれぞれ任意の処理コードを暗号化したコードであり、どれも復号処理コード 3 0 及び復号処理コード 3 1 で復号しなければ元の処理を行う処理コードに戻らない。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日	1990年 8月29日
[変更理由]	新規登録
住 所	東京都港区芝五丁目7番1号
氏 名	日本電気株式会社